CpE 645 Image Processing and Computer Vision

Prof. Hong Man

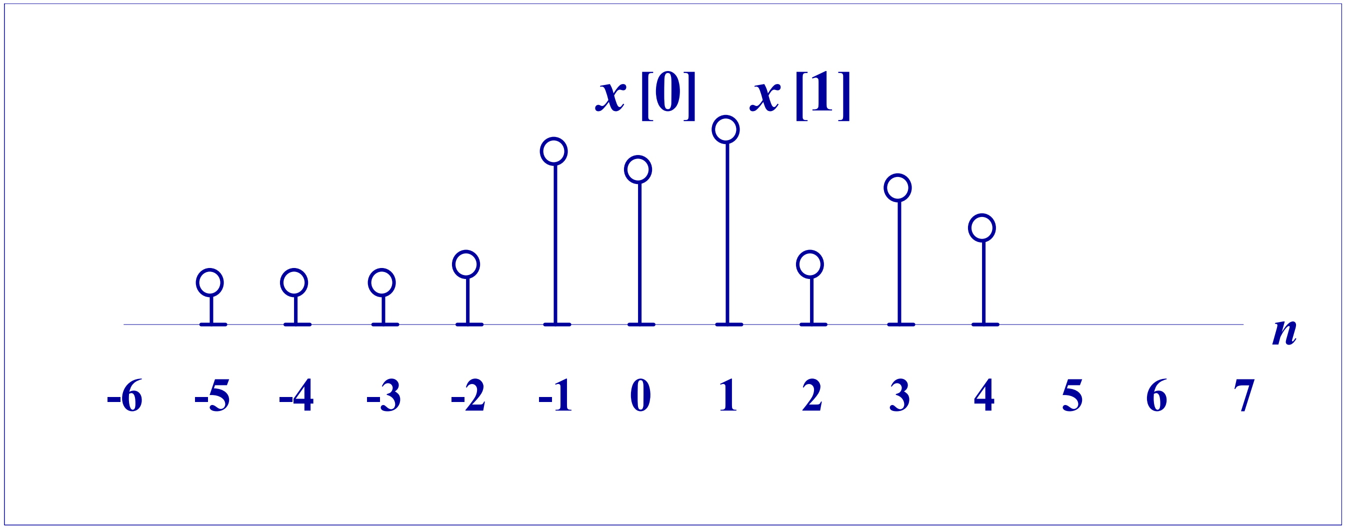
Department of Electrical and Computer Engineering
Stevens Institute of Technology

Discrete-Time Signals and Systems

- Representing signals as sequences of numbers
- Processing of those sequences

$\boldsymbol{x}=\{x[n]\}$: an indexed set of real or complex numbers.

x[n]: the nth member of the sequence.



Discrete-Time Systems

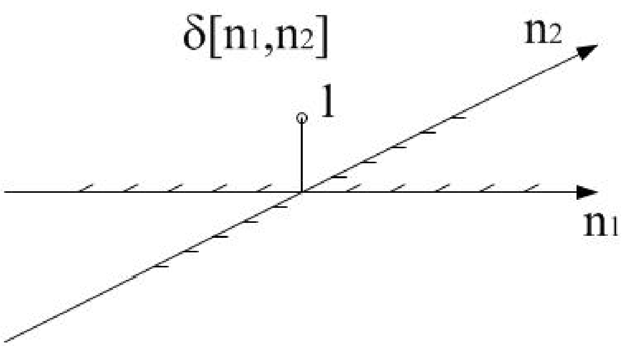
- A system is an agent for changing one sequence (input) to another (output), e.g. algorithm, program, hardware

$$x[n] \longrightarrow \boxed{\begin{array}{c} T\{\cdot\} \\ \text{System} \end{array}} \longrightarrow y[n]$$

- Processing Tasks:
  - Remove corruption (noise) from the signal (enhancement)
  - Restoration
  - Parameter extraction/estimation; modeling
  - Spectrum Estimation
  - Coding, compression, transmission
  - Analog system simulation

2-D Signals

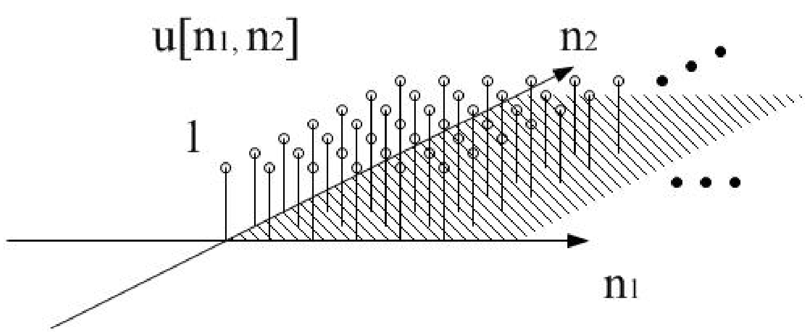
- A 2-D signal $x[n_1, n_2]$ is a function of two independent variables
- Elementary 2-D signals:
  - Unit impulse:



$$\delta[n_1, n_2] = \begin{cases} 1 & n_1 = n_2 = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$= \delta[n_1]\delta[n_2]$$

  - Unit step:



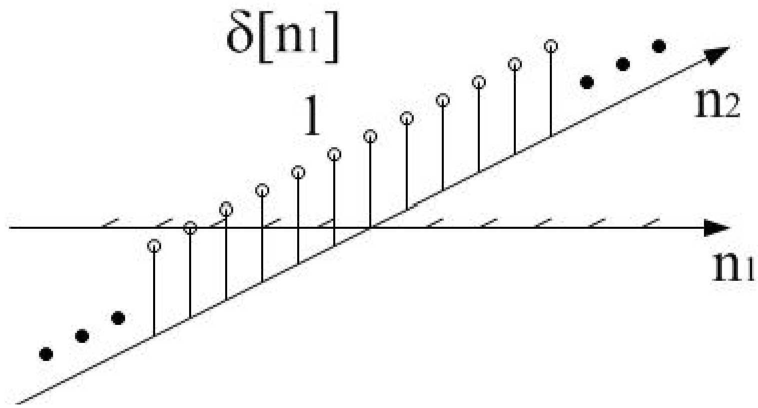
$$u[n_1, n_2] = \begin{cases} 1 & n_1, n_2 \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$= u[n_1]u[n_2]$$

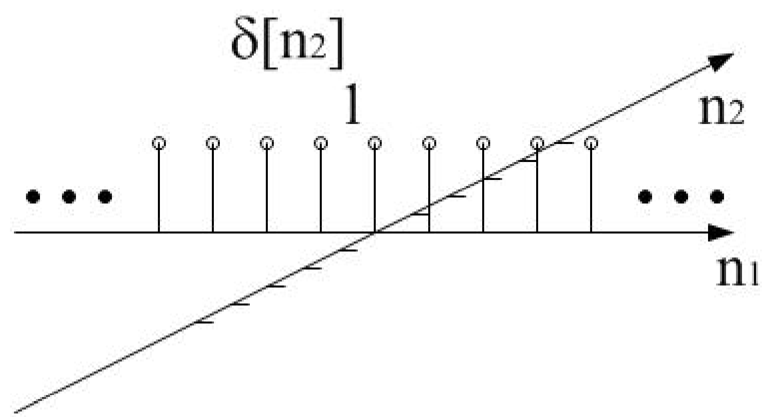
2-D LSI Systems and Signals

- Elementary 2-D signals:
  - Line impulses:

$$x_1[n_1, n_2] = \delta[n_1]1[n_2] = \delta[n_1]$$



$$x_2[n_1, n_2] = 1[n_1]\delta[n_2] = \delta[n_2]$$

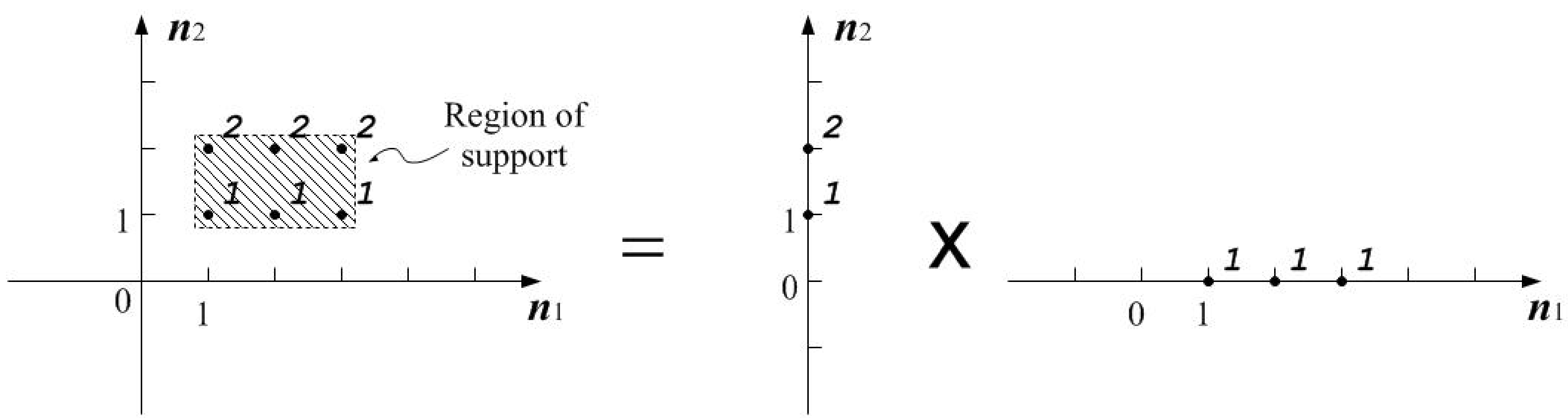


Separability

- A signal or system is separable if it (signal or impulse response) can be separated into the product of two functions $x[n_1, n_2] = x_1[n_1]x_2[n_2]$

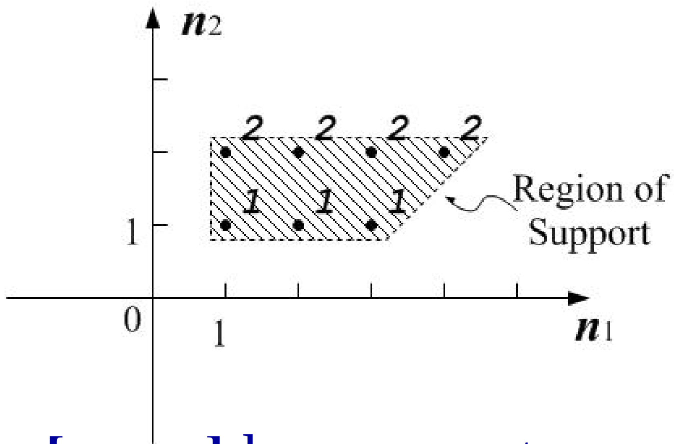
  where $x[n_1]$ only varies over n_1 and is constant over n_2, $x[n_2]$ only varies over n_2 and is constant over n_1.

- Example 1



Region of Support

- Example 2



separable ?

- A sequence $x[n_1, n_2]$ has support on a region R if

$$x[n_1, n_2] = 0, \text{ for } [n_1, n_2] \notin R$$

- In general, all practical images have finite region of support.
- A separable 2-D signal always has a rectangular region of support (necessary condition).

2-D Linear Shift Invariant System

$$x[n_1,n_2] \longrightarrow \boxed{\begin{array}{c} T\{\cdot\} \\ \text{System} \end{array}} \longrightarrow y[n_1,n_2]$$

- Additive: if $x[n_1,n_2] = (x_1[n_1,n_2] + x_2[n_1,n_2])$, then

$$y[n_1,n_2]=T\{x_1[n_1,n_2] + x_2[n_1,n_2]\}$$
$$= T\{x_1[n_1,n_2]\} + T\{x_2[n_1,n_2]\}$$

- Homogeneity: if $x[n_1,n_2]= ax_1[n_1,n_2]$, then

$$y[n_1,n_2]=T\{ax_1[n_1,n_2]\} =aT\{x_1[n_1,n_2]\}$$

- Linearity: **a** system is linear if it is both additive and homogeneous

$$T\{ax_1[n_1,n_2] + bx_2[n_1,n_2]\} = aT\{x_1[n_1,n_2]\} + bT\{x_2[n_1,n_2]\}$$

2-D Linear Shift Invariant System

- Shift invariant

$$T\{x[n_1-m_1,n_2-m_2]\}= y[n_1-m_1,n_2-m_2]$$

- If a system is both linear and shift invariant, then it is called linear and shift invariant (LSI) system.
- Exercise:

$$y[n_1,n_2]=x[n_1,n_2]\cos\omega n_1 \cos\omega n_2$$

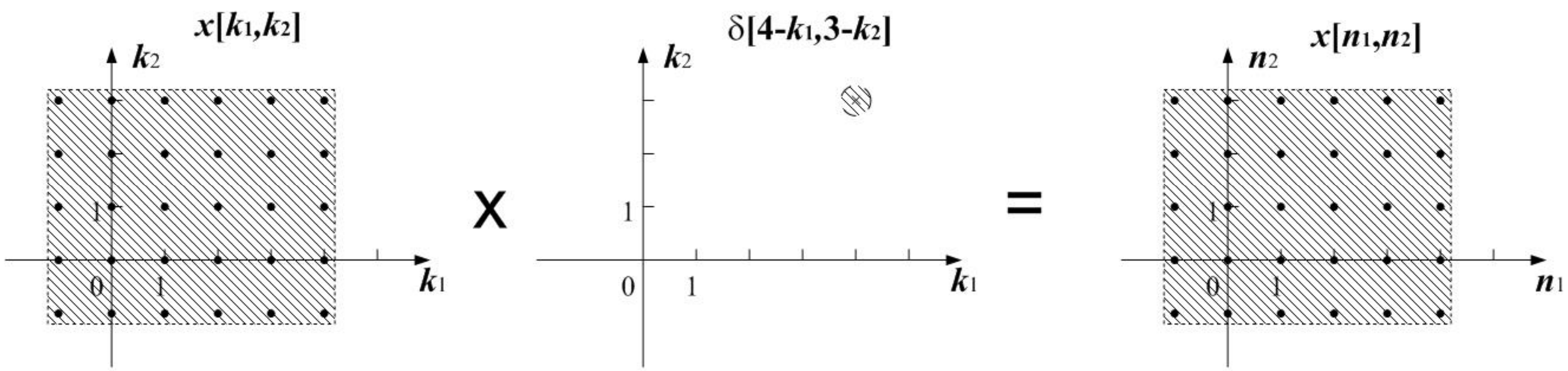
additive?	yes
homogeneous?	yes
linear?	yes
shift invariant?	no

2-D Linear Shift Invariant System

- Any 2-D signal can be decomposed into a sum of weighted and shifted 2D unit impulses as:

$$x[n_1, n_2] = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x[k_1, k_2]\, \delta]n_1 - k_1, n_2 - k_2]$$

- Example



Impulse Response

$$\delta[n_1,n_2] \longrightarrow \boxed{\begin{matrix}T\{\cdot\}\\ \text{System}\end{matrix}} \longrightarrow h[n_1,n_2]$$

- The impulse response $h[n_1,n_2]$ can fully describe a LSI system.

$$y[n_1,n_2] = T\{\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x[k_1,k_2]\delta[n_1\text{-}k_1,n_2-k_2]\}$$

$$= \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x[k_1,k_2]T\{\delta[n_1-k_1,n_2-k_2]\}$$

$$= \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x[k_1,k_2]h[n_1\text{-}k_1,n_2-k_2]$$

2-D Convolution

- 2-D convolution

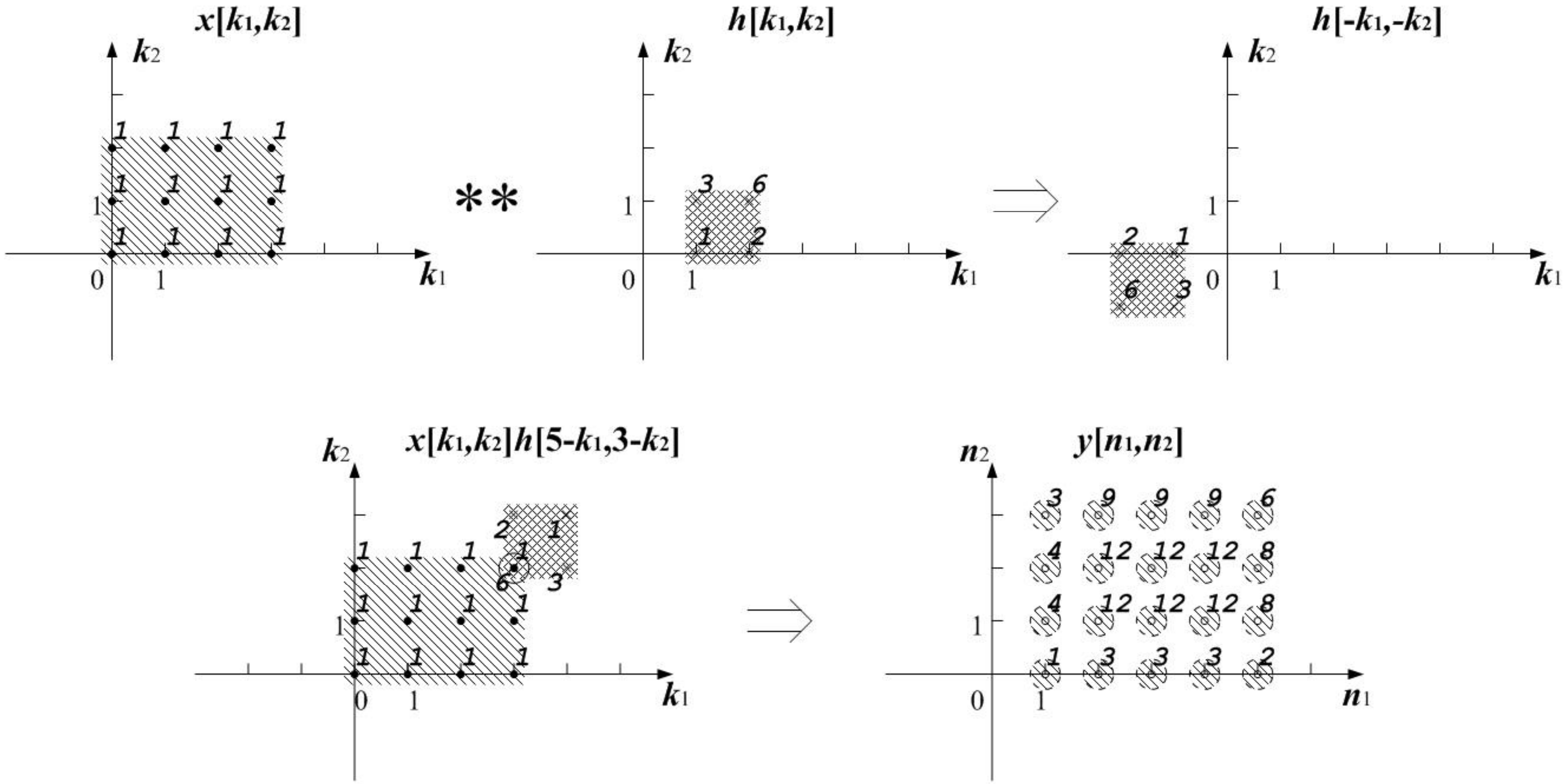
$$y[n_1,n_2] = \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} x[k_1,k_2]\, h[n_1\text{-}k_1, n_2 - k_2]$$

$$= x[n_1,n_2] ** h[n_1,n_2]$$

$$= h[n_1,n_2] ** x[n_1,n_2]$$

- Procedure:
  – Flip **h[k_1,k_2]** at both **k_1** direction and **k_2** direction to obtain **h[-k_1,-k_2].**
  – Slide (or shift) **h[-k_1,-k_2]** to obtain **h[n_1-k_1,n_2-k_2]** for all **n_1** and **n_2** (from -∞ to +∞ in theory).
  – At each particular position **[n_1, n_2]**, multiply the overlap samples, and sum up all the products to generate one output point **y[n_1, n_2].**

2-D Convolution Example



2-D Convolution Properties

- Commutative:

$$x[n_1,n_2] ** h[n_1,n_2] = h[n_1,n_2] ** x[n_1,n_2].$$

- Associative:

$$(x[n_1,n_2] ** h[n_1,n_2]) ** g[n_1,n_2] =$$

$$x[n_1,n_2] ** (h[n_1,n_2] ** g[n_1,n_2])$$

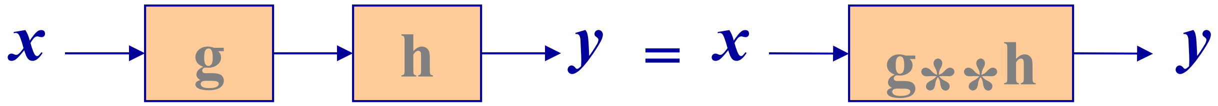
- Distributive with respect to addition:

$$x[n_1,n_2] ** (h[n_1,n_2] + g[n_1,n_2]) =$$

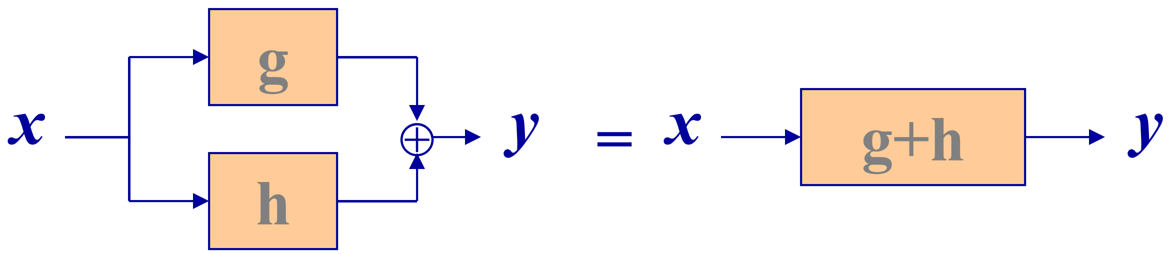
$$(x[n_1,n_2] ** h[n_1,n_2]) + (x[n_1,n_2] ** g[n_1,n_2])$$

2-D Convolution Properties

- Cascade (associative):



- Parallel (distributive):



2-D LSI System Stability

- Stability (bounded input bounded output):

$$\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}\left|h[n_1,n_2]\right| = S < \infty$$

  LSI is BIBO stable. This guarantees

$$\text{if } \left|x[n_1,n_2]\right| \le B, \text{ then } \left|y[n_1,n_2]\right| \le B', \ \ \forall[n_1,n_2]$$

  i.e. absolute summable.

- Some facts:
  - All FIR systems are BIBO stable.
  - Stability of IIR systems is usually difficult to verify.
  - An ideal LPF is unstable.

Convolution with Separable System

- For a separable 2-D LSI system, we have

$$y[n_1,n_2]=\sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty}x[k_1,k_2]h_1[n_1-k_1]h_2[n_2-k_2]$$

$$=\sum_{k_1=-\infty}^{\infty}h_1[n_1-k_1]\sum_{k_2=-\infty}^{\infty}x[k_1,k_2]h_2[n_2-k_2]$$

$$=\sum_{k_1=-\infty}^{\infty}h_1[n_1-k_1]\underline{\left(x[k_1,n_2]*h_2[n_2]\right)}$$

let $\left(x[n_1,n_2]*h_2[n_2]\right) = g[n_1,n_2]$

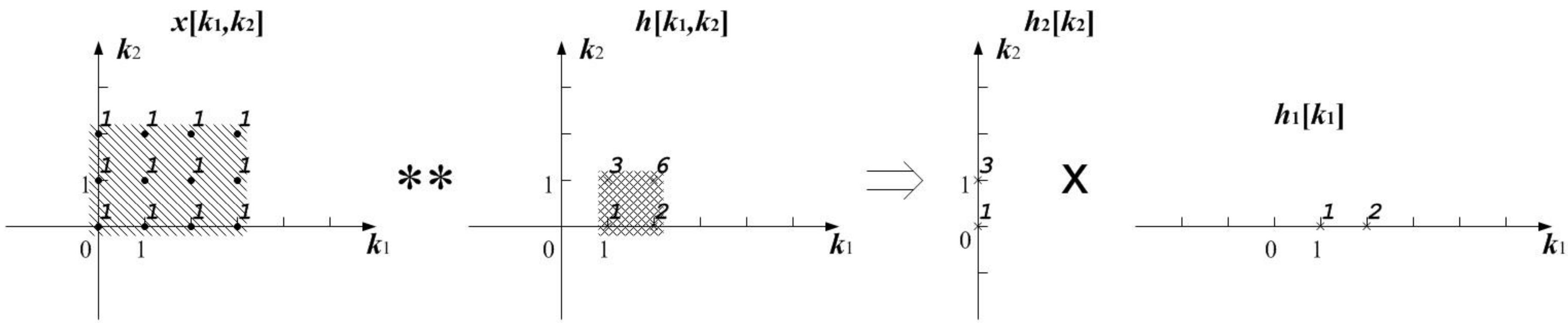
$$=\sum_{k_1=-\infty}^{\infty}h_1[n_1-k_1]g[k_1,n_2]$$

$$=h_1[n_1]*g[n_1,n_2]$$

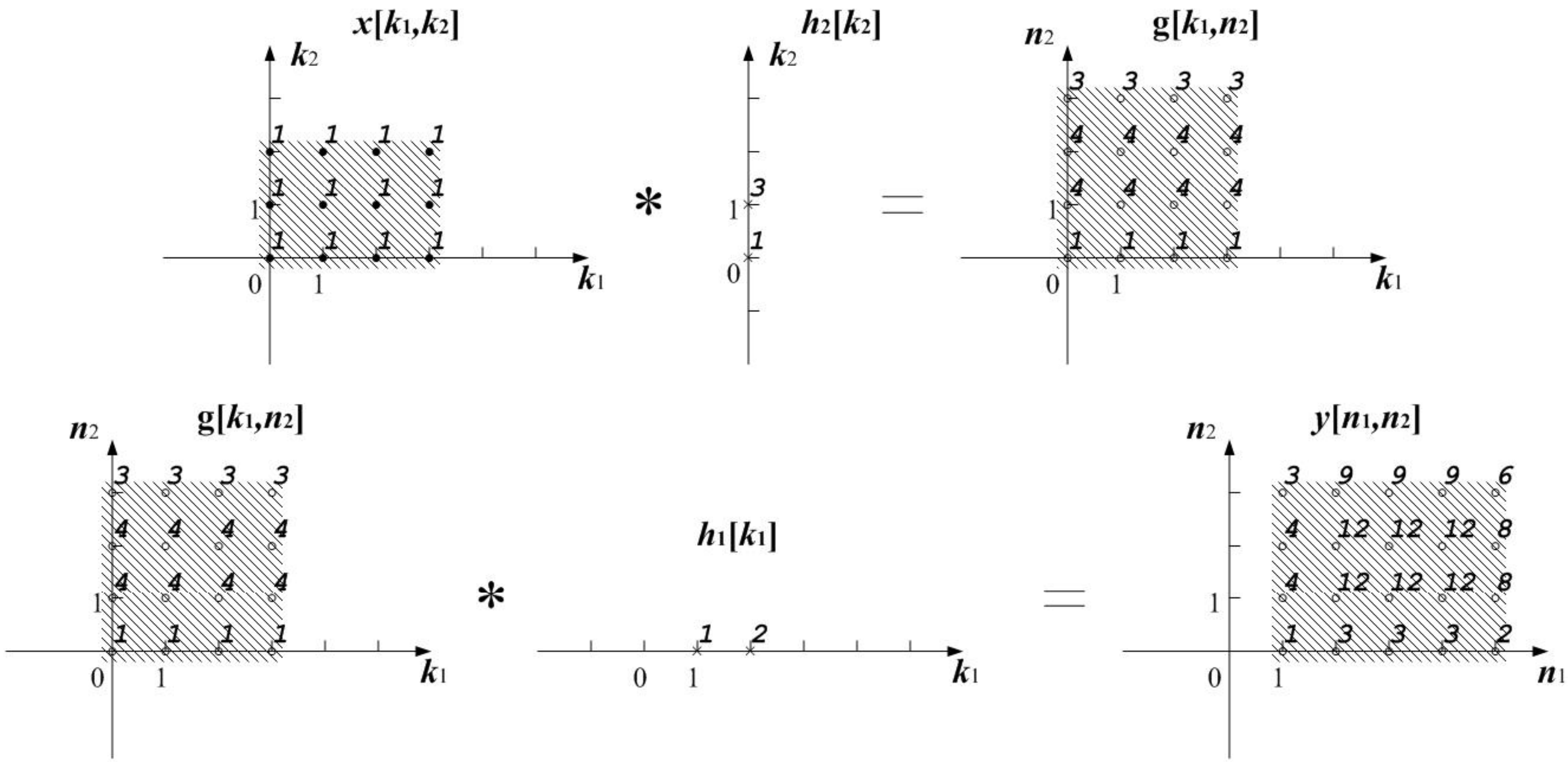
Convolution with Separable System

- Therefore for a separable 2-D LSI system, we can perform the 2-D convolution at two passes:
  - Convolute each row $\mathbf{n_1}$ of $\mathbf{x[n_1,n_2]}$ with 1-D sequence $\mathbf{h_1[n_1]}$ to obtain $\mathbf{g[n_1,n_2]}$.
  - Convolute each column $\mathbf{n_2}$ of $\mathbf{g[n_1,n_2]}$ with 1-D sequence $\mathbf{h_2[n_2]}$.
- The row-column order can be inversed, i.e.
  - Convolute each column $\mathbf{n_2}$ of $\mathbf{x[n_1,n_2]}$ with 1-D sequence $\mathbf{h_2[n_2]}$ to obtain $\mathbf{\hat{g}[n_1,n_2]}$.
  - Convolute each row $\mathbf{n_1}$ of $\mathbf{\hat{g}[n_1,n_2]}$ with 1-D sequence $\mathbf{h_1[n_1]}$.

Separable Convolution Example



Separable Convolution Example



Convolution with Separable System

- Separable system has a computational advantage in 2-D convolution:
  - Assume an image of infinite size, and a separable 2-D system with size of $\mathbf{m \times n}$.
  - To convolve this image with this system, the average number of multiplications per output sample is $\mathbf{m \times n}$.
  - If the system can be separated into two 1-D systems, with sizes of **m** and **n** respectively, then the average number of multiplications per output sample is $\mathbf{m+n}$**,** which can be much smaller than $\mathbf{m \times n}$**.**

Difference Equation Representation

- A 2-D LSI system can be generally represented in the form of a difference equation

$$y[n_1,n_2] = -\sum_{k_1}\sum_{k_2} b[k_1,k_2]\, x[n_1-k_1, n_2-k_2] + \sum_{m_1}\sum_{m_2} a[m_1,m_2]\, y[n_1-m_1, n_2-m_2]$$

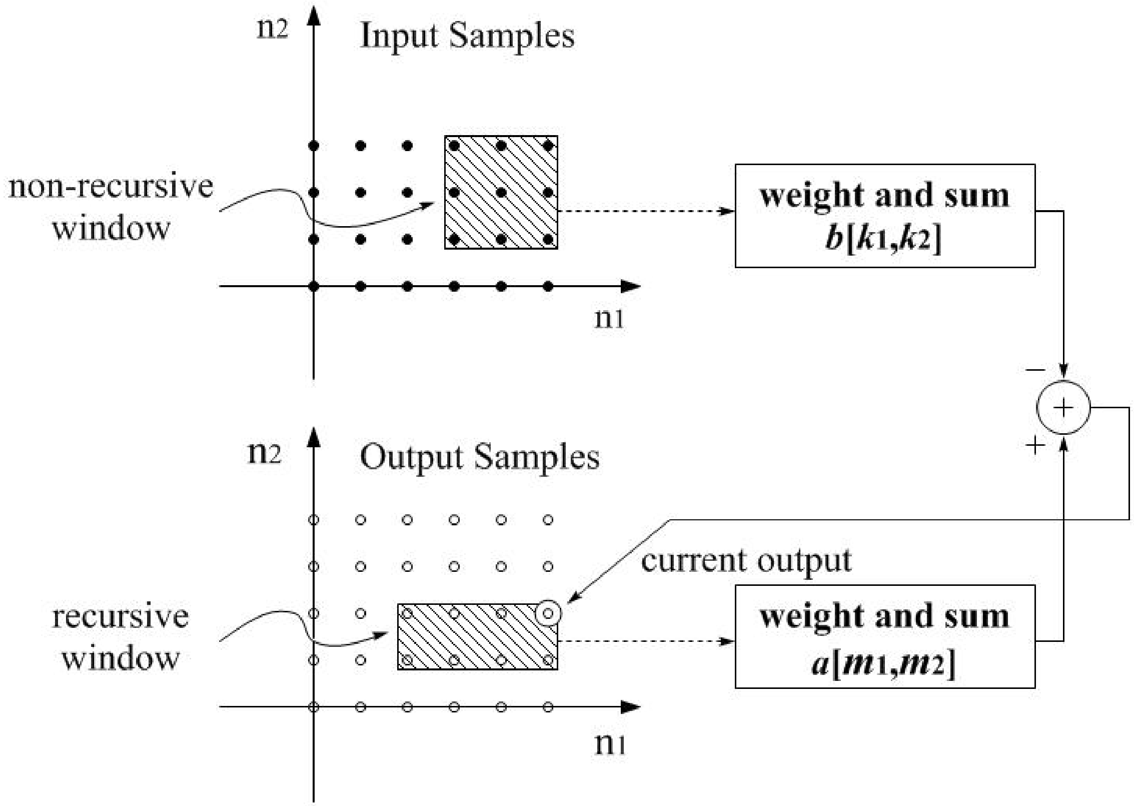
  – Coefficients $\boldsymbol{b[k_1,k_2]}$ are associated with feed-forward component of the system which only depends on system input.
  – Coefficients $\boldsymbol{a[m_1,m_2]}$ are associated with feed-back or recursive component, which depends on the other system output.

Difference Equation Representation

- If $\boldsymbol{a}[\boldsymbol{m}_1,\boldsymbol{m}_2]$ has a region of support with size of $\mathbf{M_1 \times M_2}$ the system is called a $\mathbf{(M_1 \times M_2)}$**-order** recursive system.
- A zero-order system, which contains only the $\boldsymbol{b}[\boldsymbol{k}_1,\boldsymbol{k}_2]$ coefficients, is called a finite impulse response (FIR) system.
- A non-zero order is called a infinite impulse response (IIR) system.
- For a system to be implementable, both $\boldsymbol{a}[\boldsymbol{m}_1,\boldsymbol{m}_2]$ and $\boldsymbol{b}[\boldsymbol{k}_1,\boldsymbol{k}_2]$ much have finite region of support.

Difference Equation Representation

- The implementation of an LSI system can be interpreted as two sliding windows.



Difference Equation Representation

- The "weight and sum" with input samples is just a 2-D convolution.
- The "weight and sum" with output samples requires that values for all output samples inside the recursive window to be available.
  - For recursive windows that satisfy this condition are called "recursively computable".
  - Recursive computable depends on the
    - position of current output relating to the recursive window, and
    - the scanning order of the recursive window.

Frequency Response of 2-D System

$$e^{j(\omega_1 n_1+\omega_2 n_2)} \longrightarrow \boxed{h[n_1,n_2]} \longrightarrow y[n_1,n_2]$$

$$y[n_1,n_2]=\sum_{k_1}\sum_{k_2} e^{j(\omega_1[n_1-k_1]+\omega_2[n_2-k_2])} h[k_1,k_2]$$

$$= e^{j\omega_1 n_1} e^{j\omega_2 n_2}\left[\sum_{k_1}\sum_{k_2} e^{-j[\omega_1 k_1+\omega_2 k_2]} h[k_1,k_2]\right]$$

$$= e^{j(\omega_1 n_1+\omega_2 n_2)} H(\omega_1,\omega_2)$$

$H(\omega_1, \omega_2)$ is the system frequency response to the complex sinusoid signal $e^{j(\omega_1 n_1+\omega_2 n_2)}$, where ω_1 is horizontal frequency and ω_2 is vertical frequency.

Frequency Response of 2-D System

- For an LSI system, the response of a complex sinusoid is another complex sinusoid with same frequency but modulated amplitude and phase.
- An LSI system is able to discriminate among sinusoidal signals on the basis of their frequencies

  If $|H(\omega_1, \omega_2)| \rightarrow 1$, the frequency component (ω_1, ω_2) passes.

  If $|H(\omega_1, \omega_2)| \rightarrow 0$, the frequency component (ω_1, ω_2) stops.

2-D Discrete Time Fourier Transform

- 2-D DTFT

$$X(\omega_1,\omega_2)=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}x[n_1,n_2]e^{-j(\omega_1 n_1+\omega_2 n_2)}$$

- 2-D Inverse DTFT

$$x[n_1,n_2]=\frac{1}{(2\pi)^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}X(\omega_1,\omega_2)e^{j(\omega_1 n_1+\omega_2 n_2)}d\omega_1 d\omega_2$$

- The 2-D DTFT exists whenever the sequence $\mathbf{x[n_1,n_2]}$ is absolutely summable, i.e. the frequency response of an LSI system exists only if the system is stable.

2-D DTFT Properties

Given $x[n_1, n_2]$	$\xleftrightarrow{\text{2-D DTFT}}$	$X(\omega_1, \omega_2)$
$x[n_1 - m_1, n_2 - m_2]$	$\leftrightarrow$	$e^{-j\omega_1 m_1} e^{-j\omega_2 m_2} X(\omega_1, \omega_2)$
$e^{j\alpha_1 n_1} e^{j\alpha_2 n_2} x[n_1, n_2]$	$\leftrightarrow$	$X(\omega_1 - \alpha_1, \omega_2 - \alpha_2)$
$x[-n_1, n_2]$	$\leftrightarrow$	$X(-\omega_1, \omega_2)$
$x[n_1, -n_2]$	$\leftrightarrow$	$X(\omega_1, -\omega_2)$
$\sum_{n_1} \sum_{n_2} x[n_1, n_2]$	$\leftrightarrow$	$X(0,0)$
$x[0,0]$	$\leftrightarrow$	$\frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} X(\omega_1, \omega_2) d\omega_1 d\omega_2$
$x[n_1, n_2] ** y[n_1, n_2]$	$\leftrightarrow$	$X(\omega_1, \omega_2) Y(\omega_1, \omega_2)$
$x[n_1, n_2] = x_1[n_1] x_2[n_2]$	$\leftrightarrow$	$X(\omega_1, \omega_2) = X_1(\omega_1) X_2(\omega_2)$

The Convolution Theorem

If $\quad x[n_1, n_2] = e^{j(\omega_1 n_1 + \omega_2 n_2)},$ then $\quad y[n_1, n_2] = H(\omega_1, \omega_2) e^{j(\omega_1 n_1 + \omega_2 n_2)}$

Because any signal $x[n_1,n_2]$ can be expressed as a collection (or integration) of infinite number of complex sinusoids

$$x[n_1, n_2] = \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} X(\omega_1, \omega_2) e^{j(\omega_1 n_1 + \omega_2 n_2)} d\omega_1 d\omega_2$$

$$= \lim_{\Delta\omega_1 \to 0} \lim_{\Delta\omega_2 \to 0} \frac{1}{(2\pi)^2} \sum_{k_1} \sum_{k_2} X(\Delta\omega_1, \Delta\omega_2) e^{j(k_1 \Delta\omega_1 n_1 + k_2 \Delta\omega_2 n_2)} \Delta\omega_1 \Delta\omega_2$$

Based on the principle of superposition of a linear system

$$y[n_1, n_2] = \lim_{\Delta\omega_1 \to 0} \lim_{\Delta\omega_2 \to 0} \frac{1}{(2\pi)^2} \sum_{k_1} \sum_{k_2} X(\Delta\omega_1, \Delta\omega_2) H(\omega_1, \omega_2) e^{j(k_1 \Delta\omega_1 n_1 + k_2 \Delta\omega_2 n_2)} \Delta\omega_1 \Delta\omega_2$$

$$= \frac{1}{(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} X(\omega_1, \omega_2) H(\omega_1, \omega_2) e^{j(\omega_1 n_1 + \omega_2 n_2)} d\omega_1 d\omega_2$$

Therefore $\quad Y(\omega_1, \omega_2) = X(\omega_1, \omega_2) H(\omega_1, \omega_2)$

2-D DTFT Pairs

Given $x[n_1,n_2]$	$\xleftrightarrow{\text{2-D DTFT}}$	$X(\omega_1,\omega_2)$
$\delta[n_1, n_2]$	$\leftrightarrow$	1
$\delta[n_1 - m_1, n_2 - m_2]$	$\leftrightarrow$	$e^{-j\omega_1 m_1} e^{-j\omega_2 m_2}$
$1 \ (-\infty < n_1, n_2 < \infty)$	$\leftrightarrow$	$4\pi^2\delta\left[((\omega_1))_{2\pi}, ((\omega_2))_{2\pi}\right]$
$e^{j\alpha_1 n_1} e^{j\alpha_2 n_2}$	$\leftrightarrow$	$4\pi^2\delta\left[((\omega_1 - \alpha_1))_{2\pi}, ((\omega_2 - \alpha_2))_{2\pi}\right]$
$\begin{cases} 1, & \lvert n_1 \rvert \le N_1, \lvert n_2 \rvert \le N_2 \\ 0, & \text{otherwise} \end{cases}$	$\leftrightarrow$	$\dfrac{\sin[\omega_1(N_1 + 1/2)]}{\sin(\omega_1/2)} \dfrac{\sin[\omega_2(N_2 + 1/2)]}{\sin(\omega_2/2)}$
$\dfrac{\sin \alpha_1 n_1}{\pi n_1} \dfrac{\sin \alpha_2 n_2}{\pi n_2}$	$\leftrightarrow$	$\begin{cases} 1, & \lvert\omega_1\rvert < \alpha_1, \lvert\omega_2\rvert < \alpha_2 \\ 0, & \alpha_1 < \lvert\omega_1\rvert \le \pi, \alpha_2 < \lvert\omega_2\rvert \le \pi \end{cases}$

2-D Z-Transform

- The z-transform is a generalization of the discrete time Fourier transform in the sense that it uses general complex exponential signal z^{-n} instead of complex sinusoid signal $e^{-j\omega n}$ as the basis function (eigenfunction).
- The z-transform usually provides convenient notation for analytical study of discrete-time systems.
- Comparing to DTFT

$$X(\omega_1, \omega_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x[n_1, n_2] e^{-j(\omega_1 n_1 + \omega_2 n_2)}$$

2-D z-transform is defined as

$$X(z_1, z_2) = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} x[n_1, n_2] z_1^{-n_1} z_2^{-n_2}$$

2-D Z-Transform

- The range of values for which the summation does converge is called the region of convergence (ROC).
- For 2-D signals (images) of finite size, the z-transform exists for all z_1 and z_2, except possibly at $(z_1, z_2)=0$, or $(z_1, z_2)\to\infty$.
- For 2-D signals with quadrant support, if a point (Z_1, Z_2) lies within the ROC, then all points (z_1, z_2) are in the ROC if $|z_1| \geq |Z_1|$ and $|z_2| \geq |Z_2|$
- Important pairs:

$$\delta[n_1, n_2] \leftrightarrow 1$$

$$\delta[n_1 - m_1, n_2 - m_2] \leftrightarrow z_1^{-m_1} z_2^{-m_2}$$

2-D Z-Transform

- The inverse *z*-transform is defined as

$$x[n_1, n_2] = \frac{1}{(2\pi j)^2} \oint_{C_2} \oint_{C_1} X(z_1, z_2) z_1^{n_1 - 1} z_2^{n_2 - 1} dz_1 dz_2$$

  where the contour integration is performed in a counter-clockwise direction over a closed contour C_1, C_2 in side the ROC.

- A more practical approach to compute inverse z-transform is to form a difference equation and use it to evaluate the system response to an impulse $\delta[n_1,n_2]$.

2-D Z-Transform Properties

Given $x[n_1, n_2] \xleftrightarrow{\text{2-D Z-trans}} X(z_1, z_2)$

$y[n_1, n_2] \xleftrightarrow{\text{2-D Z-trans}} Y(z_1, z_2)$

$$ax[n_1, n_2] + by[n_1, n_2] \leftrightarrow aX(z_1, z_2) + bY(z_1, z_2)$$

$$x[n_1 - m_1, n_2 - m_2] \leftrightarrow z_1^{-m_1} z_2^{-m_2} X(z_1, z_2)$$

$$x[-n_1, n_2] \leftrightarrow X(z_1^{-1}, z_2)$$

$$x[n_1, -n_2] \leftrightarrow X(z_1, z_2^{-1})$$

$$x[n_1, n_2] ** y[n_1, n_2] \leftrightarrow X(z_1, z_2) Y(z_1, z_2)$$

$$x[n_1, n_2] = x_1[n_1] x_2[n_2] \leftrightarrow X(z_1, z_2) = X_1(z_1) X_2(z_2)$$

2-D Z-Transform Examples

- Based on the properties, the conversion between the spatial domain and the z-domain is simple.
- Example 1:

$$x[n_1, n_2] = 2\delta[n_1 + 1, n_2] + \delta[n_1, n_2] + 2\delta[n_1, n_2 - 1] + 8\delta[n_1 - 2, n_2 - 3]$$

$$\Rightarrow X(z_1, z_2) = 2z_1 + 1 + 2z_2^{-1} + 8z_1^{-2}z_2^{-3}$$

2-D Z-Transform Examples

- Example 2:

$$y[n_1, n_2] = b_0 x[n_1, n_2] + b_1 x[n_1 - 1, n_2] - b_2 x[n_1, n_2 - 1]$$
$$- a_1 y[n_1 - 1, n_2] + a_2 y[n_1, n_2 - 1] + a_3 y[n_1 - 1, n_2 - 1]$$
$$\Rightarrow Y(z_1, z_2) = b_0 X(z_1, z_2) + b_1 z_1^{-1} X(z_1, z_2) - b_2 z_2^{-1} X(z_1, z_2)$$
$$- a_1 z_1^{-1} Y(z_1, z_2) + a_2 z_2^{-1} Y(z_1, z_2) + a_3 z_1^{-1} z_2^{-1} Y(z_1, z_2)$$

and the system function becomes

$$H(z_1, z_2) = \frac{Y(z_1, z_2)}{X(z_1, z_2)} = \frac{b_0 + b_1 z_1^{-1} - b_2 z_2^{-1}}{1 + a_1 z_1^{-1} - a_2 z_2^{-1} - a_3 z_1^{-1} z_2^{-1}}$$